AD-A105 232 DOUGLAS AIRCRAFT CO LONG BEACH CA F/G 20/4
COMPUTATION OF GENERAL TWO-DIMENSIONAL STEADY VISCOUS FLOWS,(U)
JUL 81 P G WILLIAMS, T CEBECI N00019-80-C-0036
UNCLASSIFIED MDC-J2248 NL

1 OF 1
AD
A105232

END
DATE
FILMED
10 81
DTIC

PREPARED UNDER
CONTRACT NO. N00019-80-C-0036
FOR
NAVAL AIR SYSTEMS COMMAND
WASHINGTON, D.C. 20361

DOUGLAS AIRCRAFT COMPANY

MCDONNELL DOUGLAS
CORPORATION

14

REPORT No. MDC-J2248

1

COMPUTATION OF GENERAL TWO-DIMENSIONAL STEADY VISCOUS FLOWS

BY

10

P. G. WILLIAMS [redacted] TUNCER CEBECI

PREPARED UNDER

15

CONTRACT No. N00019-80-C-0036

FOR

NAVAL AIR SYSTEMS COMMAND
WASHINGTON D.C. 20361

12 42

11

JUL [redacted]81

116400

TABLE OF CONTENTS

1.0 INTRODUCTION

In a previous report[1] it was shown that a code based on the biharmonic formulation of the Navier-Stokes equations with the corresponding Newtonized difference equations solved by a direct band solver could be quite competitive in situations where considerably more grid points are used in one direction than in the other, such as flow in a long channel. The main advantages of this approach are that it is straightforward and robust. Both of these terms are subjective of course.

To justify the first we may summarize the approach as follows: From derivative coefficients generated for the x and y variable grids we can immediately write down the difference equations (section 3) and it is then a simple matter to write down the Newton equations. The coefficients are stored by diagonals as required by the band solver and the righthand sides computed, then the Newton corrections are obtained by calls to the band solver. Also the uniformity of the approach used for introducing the boundary conditions in section 4 simplifies the treatment of a wide variety of boundary conditions.

By robustness we mean less likely to fail in a wide variety of situations where other methods, possibly more efficient but requiring careful tuning, may be unable to produce a solution at all. In a situation where one has an efficient well tuned method being applied to a familiar problem, it is difficult to tolerate the large storage requirements and possibly longer computer times needed by this approach. But in situations where an unfamiliar and awkward problem has to be solved quickly without too much tuning, robustness can be a very welcome property. We are mainly thicking of the avoidance of the all Reynolds number problem, which was demonstrated in reference 1. This has advantages when a solution of some sort is required for very large Reynolds numbers, and also when a solution is required for a moderate Reynolds number on a coarse grid, for example, when using h^2-extrapolation.

As discussed in reference 1, the main drawback of the approach is the magnitude of the storage requirements. Although this could be overcome by making use of direct-access storage devices, it was felt that other means of reducing the storage requirements should be explored first, especially if they may lead to a reduction in computing time as well. The present report documents progress on this in four main directions. Also in two appendices tentative work is reported which carried out an alternative to the biharmonic scheme.

The biharmonic code described in reference 1 was restricted to laminar flows and uniform grids. The code for which improvements are being considered is a more general version of that laminar code in which turbulence can be taken into account by using an eddy viscosity. The differential equations for this form of the stream function equation are derived in section 2.

With regard to the improvements to this code, we first consider the introduction of variable grids in the x and y directions. For this we need formulas for higher derivatives up to the fourth in which the grid spacing is not assumed to be uniform. These are derived in section 3. We also derive the modifications to the coefficients in the Newton matrix that this generalization will engender in section 5. This improvement is fully implemented and tested, and has been applied to the test problems considered in sections 8 and 9.

The second technique that we explore is to tailor the band solver algorithm so as to take full advantage of the considerable number of zeros that occur within the band of the matrix. The most obvious way of doing this is to perform the LU decomposition by a block-elimination scheme. This is considered in section 6, where the formulas for the forward and backward sweeps are derived. Experiments with related ADI schemes are described in Appendices A and B.

A third technique that could be used to introduce some flexibility as well as to reduce storage requirements and/or computer time is to divide the flow region into subregions and use the direct LU decomposition procedure on each of the smaller subregions. Storage requirements for the decomposed matrix for the smaller subregions will be much less at the expense of an iterative procedure to couple the regions together. If the number of subregions is fairly small, this outer iteration should converge quickly and, since the inner iterations for the subregions should be much faster, it may be possible to reduce computer time as well if the accuracy aimed at is not too great. A simple scheme for linking regions is described in section 7, but there is a potential here for more sophisticated schemes in which one could allow different grid structures in different regions, thus allowing one to take account of flow structure in a more realistic way. This could be particularly advantageous for large Reynolds numbers. The implementation of a simple version of this technique is currently being considered in order to test the effectiveness of the basic philosophy.

These first three techniques are quite general in that they can be used for whatever problem we are considering. We now restrict attention to channel flows and, in particular, the sudden expansion configuration. For large Reynolds numbers, we may neglect the upstream influence in the entrance channel and assume that the flow is fully developed right up to the entrance. Thus we may take the entrance conditions in the expanded channel as a parabolic profile over the middle section and no slip over the remainder. We may accommodate the rapid changes from these upstream conditions by taking a fine x grid close to the entrance. We may also choose a fine y grid near the walls to take account of the boundary layers there. This means that we will have a fine grid in both directions in the corners, which should take care of any problems in the corners. The chief remaining problem area is the slow approach to fully developed conditions far downstream. This can be dealt with to some extent by choosing an expanding x grid at large distances from the entrance region. However, the analysis of perturbations from fully developed conditions due to Wilson shows that the decay, although exponential, can be very slow for large Reynolds number. Thus it seems very worthwhile to make use of the asymptotic relations derived by Wilson. This is considered in section 8. We also compare its effectiveness with a simpler downstream condition, which we call the boundary-layer condition. This may be as effective for large Reynolds numbers and would be especially useful for turbulent flows since we do not have to know the fully-developed profile. Results for turbulent, two-dimensional plane channel flow are presented and discussed in section 9.

2.0 EQUATION FOR STREAM FUNCTION. INCORPORATION OF TURBULENCE MODEL.

The turbulent Navier-Stokes equations read

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{1}{\rho}\frac{\partial \tau_{xx}}{\partial x} + \frac{1}{\rho}\frac{\partial \tau_{xy}}{\partial y} - \frac{\partial}{\partial x}(\overline{u'^2}) - \frac{\partial}{\partial y}(\overline{u'v'})$$

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \frac{1}{\rho}\frac{\partial \tau_{xy}}{\partial x} + \frac{1}{\rho}\frac{\partial \tau_{yy}}{\partial y} - \frac{\partial}{\partial x}(\overline{u'v'}) - \frac{\partial}{\partial y}(\overline{v'^2}) \qquad (2.1)$$

Since $\tau_{xx} = 2\mu(\partial u/\partial x)$, $\tau_{xy} = \mu(\partial u/\partial y) + \mu(\partial v/\partial x)$, $\tau_{yy} = 2\mu(\partial v/\partial y)$, these can be written

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{\partial}{\partial x}[2\nu \frac{\partial u}{\partial x} - \overline{u'^2}] + \frac{\partial}{\partial y}[\nu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) - \overline{u'v'}]$$

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \frac{\partial}{\partial x}[\nu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) - \overline{u'v'}] + \frac{\partial}{\partial y}[2\nu \frac{\partial v}{\partial y} - \overline{v'^2}] \qquad (2.2)$$

Making the assumption that the Reynolds stresses can be modelled by an eddy-viscosity approach, we write

$$2\nu \frac{\partial u}{\partial x} - \overline{u'^2} = 2(\nu + \varepsilon_m)\frac{\partial u}{\partial x}, \qquad \nu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) - \overline{u'v'} = (\nu + \varepsilon_m)(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})$$

$$2\nu \frac{\partial v}{\partial y} - \overline{v'^2} = 2(\nu + \varepsilon_m)\frac{\partial v}{\partial y} \qquad (2.3)$$

Then if

$$b = (1 + \varepsilon_m^+), \qquad \text{where} \qquad \varepsilon_m^+ = \varepsilon_m/\nu \qquad (2.4)$$

and we introduce nondimensional variables, we obtain

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{R}\frac{\partial}{\partial x}(2b\frac{\partial u}{\partial x}) + \frac{1}{R}\frac{\partial}{\partial y}[b(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})]$$

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{R}\frac{\partial}{\partial x}[b(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})] + \frac{1}{R}\frac{\partial}{\partial y}(2b\frac{\partial v}{\partial y}) \qquad (2.5)$$

Eliminating the pressure, we obtain

$$u \frac{\partial}{\partial x}(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}) + v \frac{\partial}{\partial y}(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}) = \frac{1}{R}\left\{\frac{\partial^2}{\partial x \partial y}\left[2b(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y})\right] + (\frac{\partial^2}{\partial y^2} - \frac{\partial^2}{\partial x^2})\left[b(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})\right]\right\}$$

which in terms of the nondimensional stream function f reads

$$\frac{\partial f}{\partial y} \nabla^2 \frac{\partial f}{\partial x} - \frac{\partial f}{\partial x} \nabla^2 \frac{\partial f}{\partial y} = \frac{1}{R}\left\{ \frac{\partial^2}{\partial x \partial y}\left[4b\,\frac{\partial^2 f}{\partial x \partial y}\right] + \left(\frac{\partial^2}{\partial y^2} - \frac{\partial^2}{\partial x^2}\right)\left[b\left(\frac{\partial^2 f}{\partial y^2} - \frac{\partial^2 f}{\partial x^2}\right)\right]\right\} \qquad (2.6)$$

Differentiating out we obtain

$$4(bf_{xy})_{xy} = 4bf_{xxyy} + 4b_x f_{xyy} + 4b_y f_{xxy} + 4b_{xy} f_{xy}$$

$$[b(f_{yy} - f_{xx})]_{yy} = b\,(f_{yy} - f_{xx})_{yy} + 2b_y(f_{yy} - f_{xx})_y + b_{yy}(f_{yy} - f_{xx})$$

$$-[b(f_{yy} - f_{xx})]_{xx} = -b(f_{yy} - f_{xx})_{xx} - 2b_x(-f_{yy} + f_{xx})_x - b_{xx}(f_{yy} - f_{xx})$$

$$\therefore\ f_y\nabla^2 f_x - f_x\nabla^2 f_y = \frac{1}{R}\,[b(f_{xxxx} + 2f_{xxyy} + f_{yyyy}) + 2b_x\nabla^2 f_x + 2b_y\nabla^2 f_y + 4b_{xy} f_{xy} + (b_{yy} - b_{xx})(f_{yy} - f_{xx})]$$

Thus finally the "turbulent" stream-function equation reads

$$f_y\nabla^2 f_x - f_x\nabla^2 f_y = \frac{1}{R}\,[b\nabla^4 f + 2(b_x\nabla^2 f_x + b_y\nabla^2 f_y) + (b_{yy} - b_{xx})(f_{yy} - f_{xx}) + 4b_{xy} f_{xy}] \qquad (2.7)$$

3.0 DIFFERENCE FORMULATION WITH VARIABLE GRID

We set up a rectangular grid covering the basic rectangle $x_{min} \leq x \leq x_{max}$, $y_{min} \leq y \leq y_{max}$ with M grid lines in the x-direction and N in the y-direction. Actually we also store two exterior x and y grid values at each end because we need them in computing coefficients for derivatives actually on the boundaries. The grid spacing in each direction may be nonuniform.

Since the stream function equation involves drivatives of up to the fourth order, we require formulas for higher derivatives at the grid points of a grid with nonuniform spacing. We restrict attention to the x grid and indicate only briefly the modifications in notation needed to transform to the y grid. We can represent to second-order accuracy the 1st and 2nd derivatives in terms of three neighboring function values by locally fitting a quadratic. Thus we find that

$$f_i' = -\left[\frac{h_i/h_{i-1}}{h_i+h_{i-1}}\right] f_{i-1} + \left[\frac{h_i-h_{i-1}}{h_i \cdot h_{i-1}}\right] f_i + \left[\frac{h_{i-1}/h_i}{h_i+h_{i-1}}\right] f_{i+1} - \frac{h_i h_{i-1}}{6} f'''(x_i) + \ldots \tag{3.1}$$

$$f_i'' = \left[\frac{2/h_{i-1}}{h_i+h_{i-1}}\right] f_{i-1} - \left[\frac{2}{h_i h_{i-1}}\right] f_i + \left[\frac{2/h_i}{h_i+h_{i-1}}\right] f_{i+1} - \frac{h_i-h_{i-1}}{3} f'''(x_i) + \ldots$$

where $h_i = x_{i+1}-x_i$. From the error terms we see that the first formula is clearly second order and so is the second if the rate of change of grid spacing with x is of order unity.

For the 3rd and 4th derivatives we need to fit a quartic locally, so these formulas will involve five neighboring function values f_{i-2}, f_{i-1}, f_i, f_{i+1}, f_{i+2}. It will be convenient to adopt immediately a notation that can be transcribed directly into the Fortran code, so we write

$$\begin{aligned}
f_i' &= a_2(i)f_{i-1} + a_3(i)f_i + a_4(i)f_{i+1} \\
f_i'' &= c_2(i)f_{i-1} + c_3(i)f_i + c_4(i)f_{i+1} \qquad (3.2) \\
f_i''' &= p_1(i)f_{i-2} + p_2(i)f_{i-1} + p_3(i)f_i + p_4(i)f_{i+1} + p_5(i)f_{i+2} \\
f_i'''' &= r_1(i)f_{i-2} + r_2(i)f_{i-1} + r_3(i)f_i + r_4(i)f_{i+1} + r_5(i)f_{i+2}
\end{aligned}$$

where the coefficient notation $b_k(i)$, $d_k(i)$, $q_k(i)$, $s_k(i)$ has been reserved for the corresponding coefficients for derivatives in the y direction.

Formulas for the $a_k(i)$ and $c_k(i)$ have been given above and can easily be derived from Taylor expansions. However, for the higher derivatives the algebra is more complicated and it is convenient to work more systematically in terms of divided differences. Thus Newton's interpolating quartic reads

$$f(x) = \Delta_0 + \Delta_1(x-x_1) + \Delta_2(x-x_1)(x-x_2) + \ldots + \Delta_4(x-x_1)\ldots(x-x_4) \qquad (3.3)$$

where we have for convenience taken $i = 3$; a shift to the general local grid point can easily be made whenever required. The coefficients occurring here are the divided differences [2]

$$\Delta_n = \sum_{j=1}^{n+1} \left[\prod_{\substack{k=1 \\ k\neq j}}^{n+1} (x_j - x_k) \right]^{-1} f(x_j) \qquad (3.4)$$

To derive the higher derivative formulas we write

$$f(x) = \Delta_0 + \Delta_1(x-x_1) + \Delta_2[x^2 - (x_1+x_2)x + x_1x_2] + \Delta_3[x^3 - (x_1+x_2+x_3)x^2 + \ldots]$$
$$+ \Delta_4[x^4 - (x_1+x_2+x_3+x_4)x^3 + \ldots]$$

Then

$$f'''(x) = 6\Delta_3 + 6\Delta_4(4x-x_1-x_2-x_3-x_4), \qquad f''''(x) = 24\Delta_4,$$

and for the values at the "central" grid point x_3 we obtain

$$f'''(x_3) = 6[\Delta_3 + \Delta_4(3x_3-x_1-x_2-x_4)]$$
$$f''''(x_3) = 24\Delta_4 \qquad (3.5)$$

Thus the coefficients $r_k(3)$ are immediate from the formulas for Δ_n:

$$r_k(3) = 24 \prod_{\substack{j=1 \\ j\neq k}}^{5} (x_k-x_j)^{-1} \qquad (3.6)$$

and the $p_k(3)$ can be obtained from them by

$$p_k(3) = 6 \prod_{\substack{j=1 \\ j\neq k}}^{4} (x_k-x_j)^{-1} + \frac{1}{4}(3x_3-x_1-x_2-x_4)r_k(3) \quad \text{for} \quad k = 1, 2, 3, 4$$

and finally

$$p_5(3) = \frac{1}{4} (3x_3-x_1-x_2-x_4)r_5(3).$$

These formulas for the $p_k(3)$ can be summarized more conveniently as

$$p_k(3) = \frac{1}{4} (\overline{x} + x_k)r_k(3), \quad k = 1, 2, \ldots, 5, \tag{3.7}$$

where

$$\overline{x} = 3x_3 - x_1 - x_2 - x_4 - x_5. \tag{3.8}$$

These coefficients and the corresponding ones for the y grid are assumed to be pre-calculated and stored for each grid line. To obtain the difference equations for the grid stream function $F_{i,j}$ we first write down the formulas centered on the grid point (i,j) for the various derivatives and cross derivatives that occur, omitting for simplicity the i or j index associated with the derivative coefficients — for a_k, c_k, p_k, r_k this will always be i and for b_k, d_k, q_k, s_k it will always be j:

$$
\begin{aligned}
f_{xxxx} &= r_1F_{i-2,j} + r_2F_{i-1,j} + r_3F_{i,j} + r_4F_{i+1,j} + r_5F_{i+2,j} \\
f_{xxyy} &= c_2d_2F_{i-1,j-1} + c_2d_3F_{i-1,j} + c_2d_4F_{i-1,j+1} + \cdots + c_4d_4F_{i+1,j+1} \\
f_{yyyy} &= s_1F_{i,j-2} + s_2F_{i,j-1} + s_3F_{i,j} + s_4F_{i,j+1} + s_5F_{i,j+2} \\
f_{xxx} &= p_1F_{i-2,j} + p_2F_{i-1,j} + p_3F_{i,j} + p_4F_{i+1,j} + p_5F_{i+2,j} \\
f_{xxy} &= c_2b_2F_{i-1,j-1} + c_2b_3F_{i-1,j} + c_2b_4F_{i-1,j+1} + \cdots + c_4b_4F_{i+1,j+1} \\
f_{xyy} &= a_2d_2F_{i-1,j-1} + a_2d_3F_{i-1,j} + a_2d_4F_{i-1,j+1} + \cdots + a_4d_4F_{i+1,j+1} \\
f_{yyy} &= q_1F_{i,j-2} + q_2F_{i,j-1} + q_3F_{i,j} + q_4F_{i,j+1} + q_5F_{i,j+2}
\end{aligned}
\tag{3.9}
$$

$$f_{xx} = c_2F_{i-1,j} + c_3F_{i,j} + c_4F_{i+1,j}$$

$$f_{yy} = d_2F_{i,j-1} + d_3F_{i,j} + d_4F_{i,j+1}$$

$$f_x = a_2F_{i-1,j} + a_3F_{i,j} + a_4F_{i+1,j}$$

$$f_y = b_2F_{i,j-1} + b_3F_{i,j} + b_4F_{i,j+1}$$

Corresponding formulas can be written down for the derivatives of the eddy viscosity b which occur in the stream function equation, namely b_{xx}, b_{yy}, b_x, b_y.

If we divide through by the eddy viscosity, the stream function equation reads

$$\nabla^4 f + \frac{2b_x - Rf_y}{b} \nabla^2 f_x + \frac{2b_y + Rf_x}{b} \nabla^2 f_y + 4 \frac{b_{xy}}{b} f_{xy} + \frac{b_{xx} - b_{yy}}{b} (f_{xx} - f_{yy}) = 0,$$

which can be written for convenience in deriving the Newton equations as

$$E = 0 \qquad (3.10)$$

where

$$E = f_{xxxx} + 2f_{xxyy} + f_{yyyy} + A(f_{xxx} + f_{xyy}) + B(f_{xxy} + f_{yyy}) + Cf_{xy} + D(f_{xx} - f_{yy}) \qquad (3.11)$$

and

$$A = \frac{2b_x - Rf_y}{b}, \quad B = \frac{2b_y + Rf_x}{b}, \quad C = \frac{4b_{xy}}{b}, \quad D = \frac{b_{xx} - b_{yy}}{b} \qquad (3.12)$$

The nonlinear difference equations are obtained by replacing the f derivatives here by the variable grid formulas derived above for each interior grid point and adding boundary condition equations; see next section.

4.0 BOUNDARY CONDITIONS

We need to specify two boundary conditions on each of the four sides. These usually involve f and its normal derivatives f_n or f_{nn} and may be linear or nonlinear. Examples of typical linear boundary conditions are the following. Along an inlet boundary we usually specify u and v, i.e. f and its normal derivative. We also specify u and v on a fixed wall when the flow is laminar. Along a line of symmetry we can specify f constant and zero second normal derivative. On a downstream boundary several pairs of boundary conditions can be used, which represent the approach to fully developed conditions with varying degrees of effectiveness. Examples of linear ones are $f_n = f_{nn} = 0$, which essentially says that f does not vary with the normal coordinate n, and $f_{nn} = f_{nnn} = 0$, which we call the boundary-layer condition. A more sophisticated downstream condition is that the approach to the downstream profile is exponential (see Wilson, 1969, for the laminar case). This leads to an example of a nonlinear condition if we do not know the decay rate, as discussed below. Another example of a nonlinear condition is the law-of-the-wall condition, which may be used to replace the $f_n = 0$ condition on a fixed wall. This will be discussed further in section 9 where specific turbulent problems are treated.

The main point we wish to make about the above list of pairs of boundary conditions is that they can all be represented in the interests of uniformity as a pair of the form

$$f_0 = q(f_1,f_2), \qquad f_1 = r(f_2,f_3) \tag{4.1}$$

where the subscripts count grid lines inwards across the boundary with the boundary itself characterized by the subscript 1. Thus f_0 denotes the fictitious exterior value which is not actually stored. The functions q and r may be linear or nonlinear. In either case their partial derivatives, which will be needed in the Newton linearization, will be denoted by

$$a = \frac{\partial q}{\partial f_1}, \qquad b = \frac{\partial q}{\partial f_2}, \qquad c = \frac{\partial r}{\partial f_2}, \qquad d = \frac{\partial r}{\partial f_3} \tag{4.2}$$

Thus, for example, the linearized version of the second condition would read

$$\delta f_1 - c\delta f_2 - d\delta f_3 = r(f_2,f_3) - f_1 \tag{4.3}$$

This form is, in fact, used directly in the next section, whereas the first condition is used essentially to substitute for the unstored exterior values.

For incorporation in the system of algebraic equations, these conditions have to be particularized for each of the four sides, which we do by attaching a superscript N, S, E, W corresponding to the North, South, East or West side. Also they have to be written in terms of the local grid stream function values $F_{i,j}$, where $i = 1$ or M along the South or North side and $j = 1$ or N along the West or East side, respectively. Since the q, a, b, r, c, d may vary along the sides, they are all, in the interests of uniformity, stored as one-dimensional arrays. As an example, suppose the downstream condition $f_{nn} = f_{nnn} = 0$ is to be imposed on the North side. The first condition $f_{nn} = 0$ can be represented directly by using the variable grid derivative coefficients corresponding to the grid line $j = N$. The second condition says essentially that the second derivative is not changing, so we can represent it by imposing $f_{nn} = 0$ also on the first interior grid line $j = N-1$. The pair of conditions therefore read

$$\begin{aligned} d_2(N)F_{i,N-1} + d_3(N)F_{i,N} + d_4(N)F_{i,N+1} &= 0 \\ d_2(N-1)F_{i,N-2} + d_3(N-1)F_{i,N-1} + d_4(N-1)F_{i,N} &= 0 \end{aligned} \tag{4.4}$$

or in the notation of (4.1)

$$F_{i,N+1} = q^N(F_{i,N},F_{i,N-1}), \qquad F_{i,N} = r^N(F_{i,N-1},F_{i,N-2}) \tag{4.5}$$

where

$$\begin{aligned} q^N(F_{i,N},F_{i,N-1}) &= a^N F_{i,N} + b^N F_{i,N-1} \\ r^N(F_{i,N-1},F_{i,N-2}) &= c^N F_{i,N-1} + d^N F_{i,N-2} \end{aligned} \tag{4.6}$$

with

$$\begin{aligned} a^N &= -d_3(N)/d_4(N), \qquad & c^N &= -d_3(N-1)/d_4(N-1) \\ b^N &= -d_2(N)/d_4(N), \qquad & d^N &= -d_2(N-1)/d_4(N-1) \end{aligned} \tag{4.7}$$

Details for the other conditions will be given where they actually arise.

5.0 NEWTON MATRIX AND SOLUTION BY BAND SOLVER

Let $\underset{\sim}{F}$ be the vector of MN unknown grid stream function values F_{ij}, $i = 1, \ldots, M$, $j = 1, \ldots, N$, i.e. the interior values plus the boundary values, but not including the exterior values. We obtain MN equations for these unknowns by writing down the grid stream function equation for all the interior grid points ($MN - 2M - 2N + 4$ equations) with exterior values assumed to be expressed in terms of the F_{ij} by means of the first boundary condition along each side, adding $2(M - 2) + 2(N - 2)$ equations representing the second boundary condition on each side and finally adding four equations for the corners. The last are obtained by selecting the second boundary condition from either of the two sides meeting at the corner. In our code we optionally choose these sides to be the North and South sides.

Let $\beta_{ij}(\underset{\sim}{F})$, $i = 1, \ldots, M$, $j = 1, \ldots, N$ be functions defined so that the equations

$$\beta_{ij}(\underset{\sim}{F}) = 0, \qquad i = 1, \ldots, M, \quad j = 1, \ldots, N \tag{5.1}$$

represent the above MN equations. Then with Newtons method, if $\underset{\sim}{F}$ does not satisfy (5.1) accurately enough, it is corrected to $\underset{\sim}{F} + \underset{\sim}{\phi}$ where $\underset{\sim}{\phi}$ satisfies

$$\sum_{|\mu|+|\nu|\leq 2} \alpha_{\mu}^{\nu}\, \phi_{i+\mu,j+\nu} = -\beta_{ij}(\underset{\sim}{F})$$

the α_{μ}^{ν} being the partial derivatives of β_{ij} with respect to the $F_{k\ell}$ that it depends on. From (3.9), (3.11), (3.12) we find that with the abbreviations

$$g_x = R(f_{xxx} + f_{xyy})/b, \qquad g_y = R(f_{xxy} + f_{yyy})/b$$

we can write down immediately

α_{-2}^{0}: $r_1 + Ap_1$

α_{2}^{0} : $r_5 + Ap_5$

α_{0}^{-2}: $s_1 + Bq_1$

α_{0}^{2} : $s_5 + Ba_5$

$$\alpha_{-1}^{-1}: \quad a_2(Ad_2 + Cb_2) + c_2(2d_2 + Bd_2)$$

$$\alpha_{1}^{-1}: \quad a_4(Ad_2 + Cb_2) + c_4(2d_2 + Bd_2)$$

$$\alpha_{-1}^{1}: \quad a_2(Ad_4 + Cb_4) + c_2(2d_4 + Bd_4)$$

$$\alpha_{1}^{1}: \quad a_4(Ad_4 + Cd_4) + c_4(2d_4 + Bd_4)$$

$$\alpha_{-1}^{o}: \quad a_2(Ad_3 + Cb_3 + g_y) + c_2(2d_3 + Bd_3 + D) + r_2 + Ap_2 \tag{5.2}$$

$$\alpha_{1}^{o}: \quad a_4(Ad_3 + Cb_3 + g_y) + c_4(2d_3 + Bd_3 + D) + r_4 + Ap_4$$

$$\alpha_{o}^{-1}: \quad b_2(Bc_3 + Ca_3 - g_x) + d_2(2c_3 + Aa_3 - D) + s_2 + Bq_2$$

$$\alpha_{o}^{1}: \quad b_4(Bc_3 + Ca_3 - g_x) + d_4(2c_3 + Aa_3 - D) + s_4 + Bq_4$$

$$\alpha_{o}^{o}: \quad a_3(Ad_3 + Cb_3 + g_y) + c_3(2d_3 + Bb_3 + D) + r_3 + Ap_3$$

$$- d_3D - b_3g_x + s_3 + Bq_3$$

When $i = 2$ or $M - 1$ or $j = 2$ or $N - 1$, the α_o^2, α_2^o, α_o^{-2} or α_{-2}^o would multiply ϕ's corresponding to exterior values, so modifications must be made to take into account the relevant first boundary conditions. For example, if $i = M\text{-}1$ and $3 \leq j \leq N\text{-}2$, as in Fig. 1 the relevant first boundary condition would be of the form

$$F_{M+1,j} = q^E(F_{M,j}, F_{M-1,j})$$

with

$$\frac{\partial q^E}{\partial F_{M,j}} = a^E(j), \qquad \frac{\partial q^E}{\partial F_{M-1,j}} = b^E(j)$$

The derivatives of β_{ij} with respect to $F_{M,j}$ and $F_{M-1,j}$ then becomes

$$\alpha_1^o(i,j) + a^E(j)\alpha_2^o(i,j) \qquad \text{and} \qquad \alpha_o^o(i,j) + b^E(j)\alpha_2^o(i,j)$$

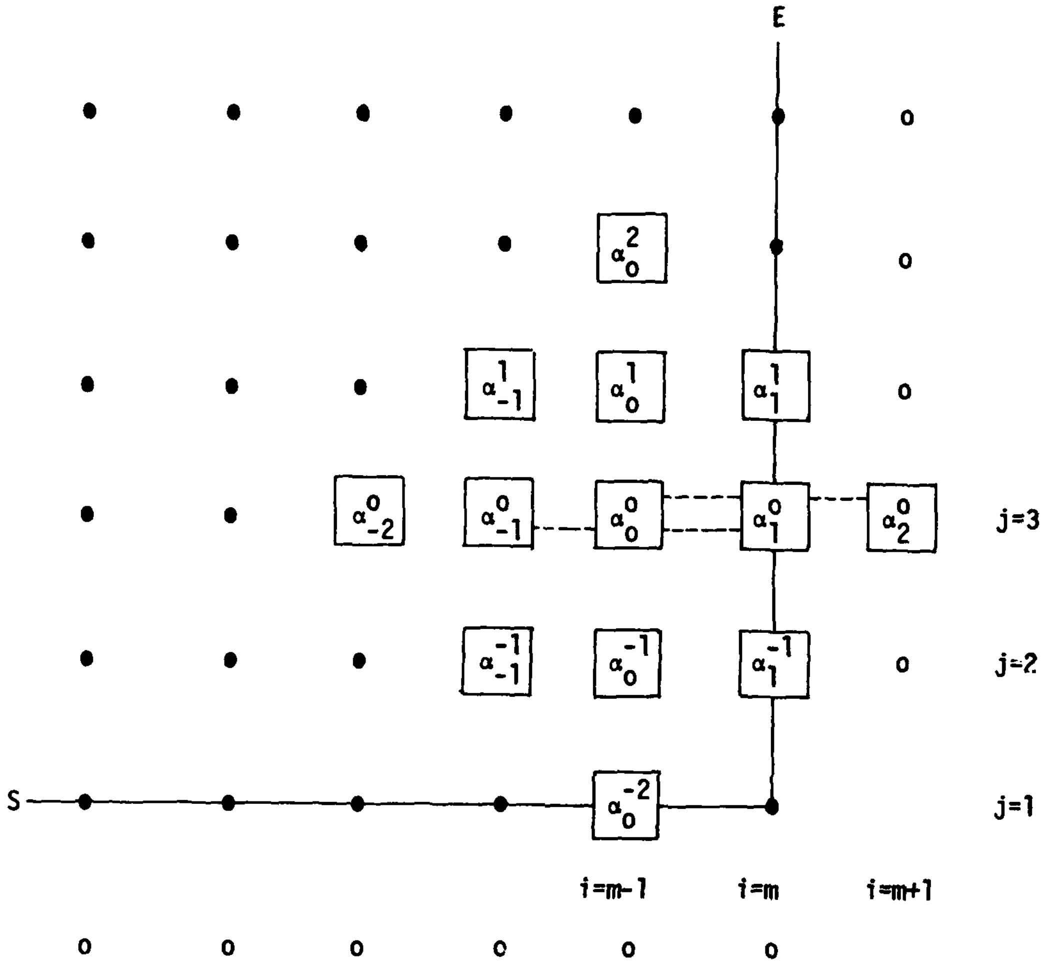

Figure 1. A typical computational molecule at a point where an exterior unstored grid value is used. The dotted lines indicate the grid values connected by the two boundary equations (4.5). Stored and unstored grid values in the calculation are indicated by ● and o.

Thus α_2^o is not used in the matrix, but is used to modify α_1^o and α_o^o. Similar modifications must be made near the other boundaries. For points neighboring both boundaries, two exterior values are involved so modifications must be made for each; thus α_o^o will then be modified twice.

The Newton equations for the second boundary conditions are added as appropriate and have the form, for example, of

$$\phi_{M,j} - c^E\phi_{M-1,j} - d^E\phi_{M-2,j} = r^E(F_{M-1,j}, F_{M-2,j}) - F_{M,j}$$

when $i = M$.

There are various ways available for the solution of the linear system formed by these MN equations. Several block elimination schemes can be formulated and also iterative schemes, including those of ADI type. If one wishes to take advantage of the chord method, i.e. the simplified Newton method in which only the right-hand sides are updated at each iteration, a scheme in which an LU decomposition is performed and stored would seem to be preferable. Block elimination schemes can be designed to do this, but because of its potentially greater stability, espeically for large Reynolds numbers, we chose to explore first the practical application of a standard band solver with partial pivoting for stability (this may be especially important if we work entirely in IBM single precision).

The standard way of organizing the equations as a banded system is to order the unknowns by rows, i.e. in the order $\phi_{1,1}, \ldots, \phi_{m,1}, \phi_{1,2}, \cdots$ The matrix elements for the system (5.2) then has the appearance shown in Fig. 2. To avoid confusion the suffices attached to the α's are shown on one row only, and the modifications to the α's due to the first boundary conditions are only indicated by position. Thus those needing modification at the West and East boundaries are shown with a bar, those at the North and South by a hat, and those at both by a bar and a hat.

For use in the band solver the diagonals have to be stored as the columns of a rectangular array D, say, so they are numbered from 1 to $\ell_4 = 4M+1$. The main diagonals of the other individual blocks also have salient locations, which we therefore denote by $\ell_1, \ell_2, \ell_3,$ so that other diagonals can be

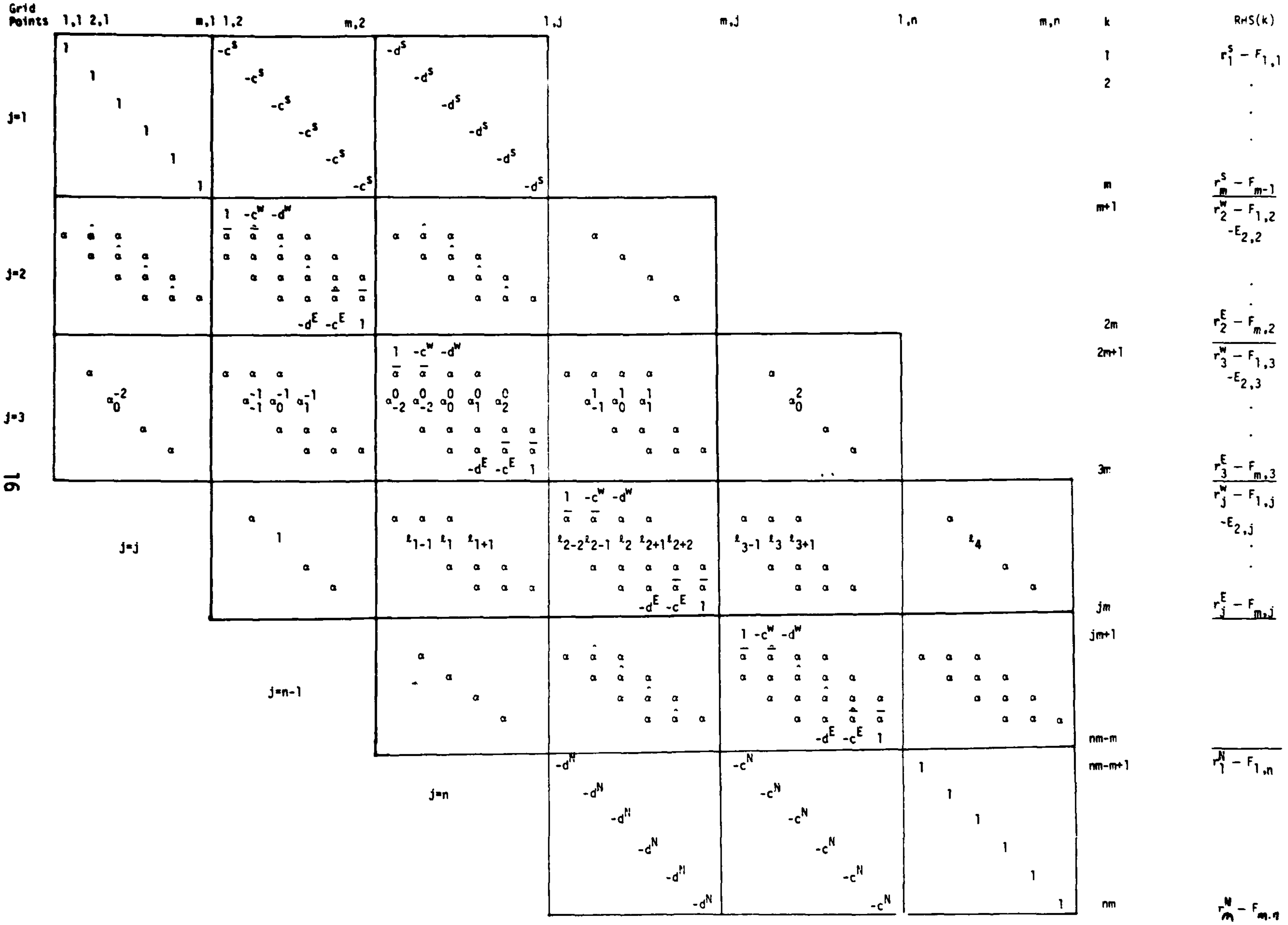

Figure 2. Newton Matrix for Biharmonic Formulation

referred to them as indicated in a typical row in Fig. 2. An equation count k is also shown in relation to the successive blocks, together with the right-hand sides.

Since the coefficients α_{μ}^{ν} are typically of order (grid size)$^{-4}$, they are much larger than the coefficients in the boundary equations. The solver therefore interchanges them in pivoting for size, which causes loss of accuracy. Strictly we should scale all the equations so that they are all fairly uniform. However, we find that the natural scaling of the stream function equations is adequate provided we scale up the boundary equations sufficiently so that they dominate and are not interchanged.

A discussion of storage estimates and operation counts, as well as iteration strategies was given in reference 1 and need not be repeated here, apart from specific details given in the examples.

6.0 QUIN-BLOCK-DIAGONAL SOLVER

A suite of subroutines has been written and tested for the LU decomposition and solution of the quin-block-diagonal system arising in the Newton iteration scheme. The system has the form

$$\begin{bmatrix} C_1 & D_1 & E_1 & & \\ B_2 & C_2 & D_2 & E_2 & \\ A_j & B_j & C_j & D_j & E_j \\ & A & B & C & D \\ & & A_n & B_n & C_n \end{bmatrix} \begin{bmatrix} \underset{\sim}{\phi}_1 \\ \underset{\sim}{\phi}_2 \\ \cdot \\ \cdot \\ \underset{\sim}{\phi}_n \end{bmatrix} = \begin{bmatrix} C_1^L & & & & \\ B_2^1 & C_2^L & & & \\ A'_j & B'_j & C_j^L & & \\ & A' & B' & C^L & \\ & & A'_n & B'_n & C_n^L \end{bmatrix} \begin{bmatrix} \underset{\sim}{\theta}_1 \\ \underset{\sim}{\theta}_2 \\ \cdot \\ \cdot \\ \underset{\sim}{\theta}_n \end{bmatrix} = \begin{bmatrix} \underset{\sim}{r}_1 \\ \underset{\sim}{r}_2 \\ \cdot \\ \cdot \\ \underset{\sim}{r}_n \end{bmatrix} ,$$

$$\begin{bmatrix} C_1^U & D'_1 & E'_1 & & \\ & C_2^U & D'_2 & E'_2 & \\ & & C^U & D' & E' \\ & & & C^U & D' \\ & & & & C_n^U \end{bmatrix} \begin{bmatrix} \underset{\sim}{\phi}_1 \\ \underset{\sim}{\phi}_2 \\ \cdot \\ \cdot \\ \underset{\sim}{\phi}_n \end{bmatrix} = \begin{bmatrix} \underset{\sim}{\theta}_1 \\ \underset{\sim}{\theta}_2 \\ \cdot \\ \cdot \\ \underset{\sim}{\theta}_n \end{bmatrix}$$

where the blocks are $m \times m$ square matrices. Here we have chosen the safer form of decomposition in which the factors are exactly triangular. The diagonal blocks C_j of the left-hand factor are unit lower triangular.

The recurrence relations for the LU decomposition used in subroutine LUBLO5 read

1. $A'_j \, C^U_{j-2} = A_j \quad (3 \leq j \leq n)$

2. $B'_j \, C^U_{j-1} = B_j - A'_j \, D'_{j-2} \quad (2 \leq j \leq n)$

3. $C^L_j \, C^U_j = P_j \, (C_j - B'_j \, D'_{j-1} - A'_j \, E'_{j-2}) \quad (1 \leq j \leq n)$

4. $C^L_j \, D'_j = P_j (D_j - B'_j \, E'_{j-1}) \quad (1 \leq j \leq n-1)$

5. $C^L_j \, E'_j = P_j \, E_j \quad (1 \leq j \leq n-2)$

where the P_j are permutation matrices introduced by the interchanges in the LU decomposition of C_j.

A subroutine SP has been written for subtracting the products involved in (2), (3), (4), a subroutine TS for the transposed forward substitutions in (1) and (2), and a subroutine FS for the forward substitutions in (4) and (5).

A subroutine SOBLE uses the decomposition to solve the block equations. The recurrence relations involved in the foward and backward substitutions read

Forward: $C^L_1\underset{\sim}{\theta}_1 = P_1\underset{\sim}{r}_1$, $C^L_2\underset{\sim}{\theta}_2 = P_2(r_2 - B'_2\underset{\sim}{\theta}_1)$

6. $C^L_j\underset{\sim}{\theta}_j = P_j(\underset{\sim}{r}_j - B'_j\underset{\sim}{\theta}_{j-1} - A'_j\underset{\sim}{\theta}_{j-2})$ $(3 \le j \le n)$

Backward: $C^U_n\underset{\sim}{\phi}_n = \underset{\sim}{\theta}_n$, $C^U_{n-1}\underset{\sim}{\phi}_{n-1} = \underset{\sim}{\theta}_{n-1} - D'_{n-1}\underset{\sim}{\theta}_n$

7. $C^U_j\underset{\sim}{\phi}_j = \underset{\sim}{\theta}_j - D'_j\underset{\sim}{\theta}_{j+1} - E'_j\underset{\sim}{\theta}_{j+2}$ $(1 \le j \le n-2)$

A vector back substitution subroutine BS was written for solving the equation in the backward sweep. Also it turned out to be convenient to write a vector subroutine S2TV for subtracting off the two transformed vectors which occur in (6) and (7).

The above subroutines have all been tested. Some significant loss in accuracy was observed in one test, but this is believed to be due to the particular test matrix chosen being somewhat ill-conditioned. Moderate loss in accuracy was also observed in another test where the matrix was of biharmonic form, so it seems that further investigation is needed before the use of a block solver is implemented in the present code. Also, in the present form the saving in storage is not substantial. Further savings could be made if a more sophisticated storage scheme was adopted, but it might be simpler to try the alternative pseudo-LU decomposition.

7.0 ITERATION BETWEEN SUB-REGIONS

Another scheme for reducing the storage requirements is to divide the flow region into sub-regions and use the direct LU decomposition procedure on each sub-region in turn. Once the local decomposition has been performed, it would seem worthwhile to iterate the simplified Newton (chord) procedure several times before proceeding to the next sub-region. The problem of connecting one region to the next can be solved very simply in the case when the grid structure is the same for all sub-regions: we merely use the band matrix for the whole region to provide the coefficients for the connecting relations between values on either side of the dividing line. Of course the remaining band sub-matrices provide the band matrices for each sub-region. This is illustrated in Figure 3 for the case of three sub-regions. The scheme would apply for any very long band matrix, but in the present biharmonic context each division would normally be chosen to correspond to the end of a grid line and the overlap vectors y_k, z_k would then correspond to the two neighboring grid lines on the other side of the boundary. The iterative procedure we would use would be essentially a block Gauss-Seidel, which for the case represented in the diagram would be written

$$A_1x_1^{(i)} = r_1 - C_1x_2^{(i-1)}$$

$$A_2x_2^{(i)} = r_2 - C_2x_3^{(i-1)} - B_2x_1^{(i)}$$

$$A_3x_3^{(i)} = r_3 - B_3x_2^{(i)}$$

where i is the iteration count and $x_2^{(i-1)}$, $x_3^{(i-1)}$ are known from the previous iteration. Note that because of the structure of the matrices B_k, C_k only the overlap parts of the vectors $x_k^{(i-1)}$ are involved. If necessary, an acceleration parameter could be introduced to accelerate convergence.

Various refinements and extensions to this basic scheme can be envisaged. For example, if the sub-regions are chosen so that within each the number of unknowns in one direction is more than in the other they should be ordered so that the M in the estimate $8M^3N$ for the LU operation count should be less than N. A generalization that would require rather more effort to implement but which should ultimately be very effective is to reformulate the

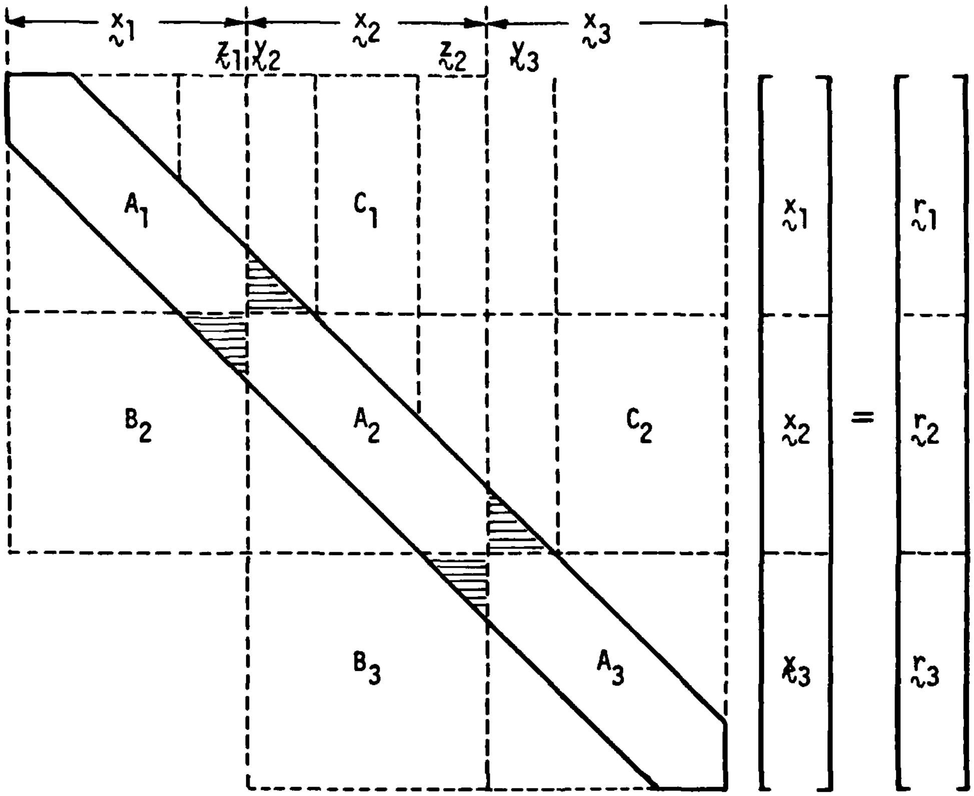

Figure 3. Newton matrix for case of three subregions.

matching conditions at the sub-region boundaries in such a way that different grid structures could be used in each sub-region. This would involve interpolation and perhaps reformulation in terms of boundary derivatives, although one could probably get away with fairly minor modifications to the present variable grid code if one restricted attention to the situation where the boundary between regions is entirely in the fluid, i.e. no part of it also has boundary conditions specified. As a fairly typical example we may consider the expanding channel of Figure 4, where the flow region is divided into three sub-regions I, II, III. The simple situation envisaged exists for the boundary between II and III and also for that between I and II looked at from inside I, but not from inside II.

We therefore consider first solving for region III assuming values in II are known from a previous iteration. We imagine the III-grid extended into region II for two y-grid lines. The x-grid lines do not match so we interpolate to obtain known values on the two grid lines outside of III. Since these are known, the Newton corrections for these values are all zero, so we can merely truncate the Newton matrix at the appropriate point and not complete it with boundary-condition equations and modifications. The code changes required for this is relatively straightforward, so we are currently considering implementing them in order to assess the effectiveness of the basic philosophy.

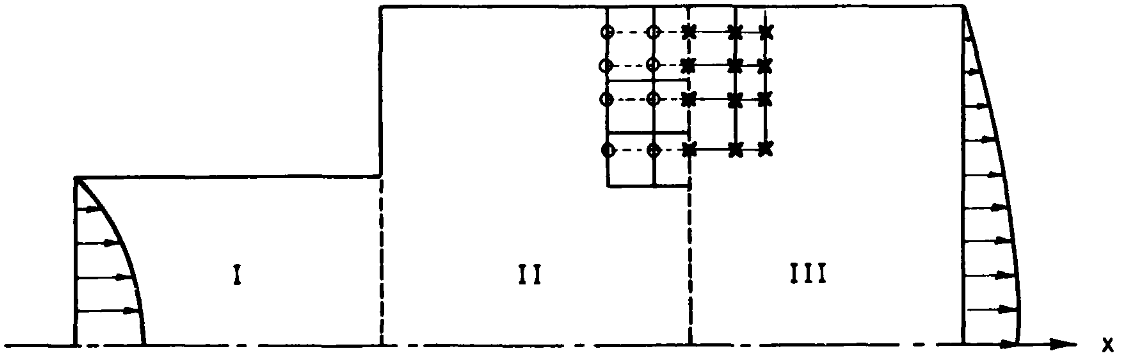

Figure 4. Expanding channel divided into subregions.

It is, of course, entirely feasible to deal with the more complicated situation, but since more sophisticated code changes would be required, it seems reasonable to hold this extension in abeyance for the immediate future, pending the above assessment.

8.0 LAMINAR FLOW APPLICATIONS

We first tested the variable grid code on the idealized channel flow problem with uniform parallel inlet velocity already calculated by the uniform grid code[1]. We omit details of the application of the code since only minor modifications were needed from those given in reference 1. It was clear that the singularities at the corner could be dealt with more satisfactorily by concentrating grid points near them, but since this situation is somewhat idealized, it was decided not to put too much effort into producing accurate results over a large Reynolds number range for this problem, but to transfer attention to the more realistic problem of the sudden expansion channel. Suffice it to say that some experimentation was performed on choice of grid structure and that the original choice of geometric grid with step $h_i = h_1 k^{i-1}$ $(k > 1)$ was ultimately abandoned in favor of a quartic grid in which the step varied cubically, the maximum and minimum of the cubic being at the boundaries, the center of the channel and the wall for the transverse grid and at the last station and the inlet for the downstream grid. This meant that for the transverse grid the step became large but fairly uniform far downstream, which is appropriate for the exponential decay.

We turn now to the sudden expansion case and again take x across the channel and y along the channel as in Fig. 5 with $x = 0$ on the wall and $x = 1/2$ on the centerline. We assume a fully developed parabolic profile at the inlet over a central opening at $y = 0$. This starts at $x = a$, where we have taken $a = 1/3$ but it can be varied. The choice of $a = 1/3$ was made because accurate experimental results are available[4]. The Reynolds number quoted for these results was $Re = 56$ and was based on the maximum velocity of a profile slightly upstream of the inlet. Assuming that this was a fully-developed profile, the corresponding Reynolds number in our notation would be $R = 41.3$. The boundary conditions on $y - 0$ are

$$f = F_0(x), \qquad f_y = 0$$

where

$$F_0(x) = \begin{cases} 0 & \text{if} \quad 0 \leq x \leq a \\ -\frac{1}{4}\left(3 - \frac{x-a}{\frac{1}{2}-a}\right)\left(\frac{x-a}{\frac{1}{2}-a}\right)^2 & \text{if} \quad a < x \leq 1/2 \end{cases}$$

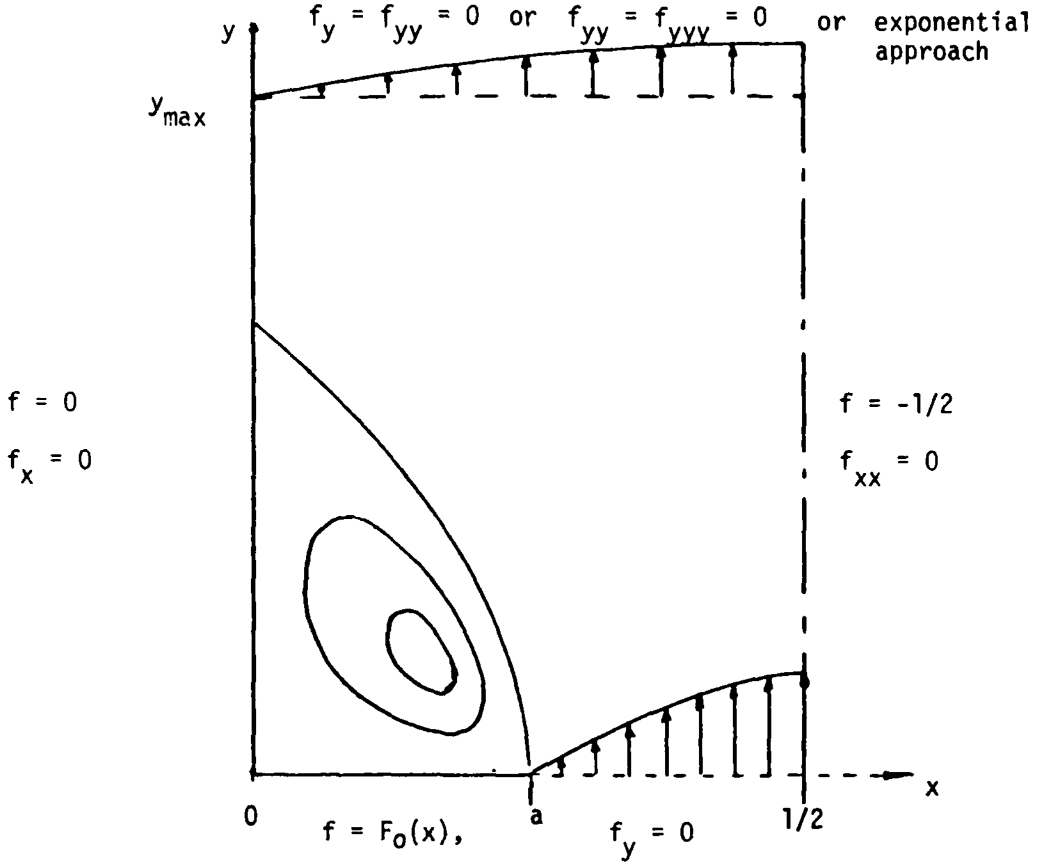

Figure 5. Boundary conditions for rapid expansion channel

The other boundary conditions are as in Fig. 5, where the several alternatives that we have used for the downstream boundary condition are shown. If the variable grid derivative coefficients are used, the table of coefficients for the boundary conditions becomes, for example;

Side	q	a	b	r	c	d
W	0	$-\frac{a_3(1)}{a_2(1)}$	$-\frac{a_4(1)}{a_2(1)}$	0	0	0
E	0	$-\frac{c_3(M)}{c_4(M)}$	$-\frac{c_2(M)}{c_4(M)}$	$-\frac{1}{2}$	0	0
S	0	$-\frac{b_3(1)}{b_2(1)}$	$-\frac{b_4(1)}{b_2(1)}$	$F_o(x)$	0	0
N	q^N	$-\frac{d_3(N)}{d_4(N)}$	$-\frac{d_2(N)}{d_4(N)}$	r^N	$-\frac{d_3(N-1)}{d_4(N-1)}$	$-\frac{d_2(N-1)}{d_4(N-1)}$

where q^N and r^N are given by (4.6), if we choose the downstream condition $f_{yy} = f_{yyy} = 0$ as represented by (4.4). This condition we call the boundary-layer condition since it implies that the streamwise second and third derivatives of the streamwise velocity are small near the downstream boundary. We compare its affectiveness with that of the following nonlinear downstream condition which models the exponential approach to the Poiseuille profile investigated by Wilson[3].

If we assume that the decay into the fully developed Poiseuille profile $f_p(x)$ as $y \to \infty$ is exponential, so that we can write for the stream function $f(x,y)$

$$f(x,y) \sim f_p(x) + e^{-\alpha y}\phi(x) \tag{8.1}$$

it can be shown (Wilson[3]) that α and $\phi(x)$, which depend on R, satisfy an equation similar to the Orr-Sommerfeld equation, and Wilson has obtained an approximate formula for α when R is large. When R is not large enough, so that we do not know α accurately, we can impose the asymptotic condition (1) by using nonlinear boundary conditions obtained by eliminating α, as shown in the last progress report. If y_N is the last y-station considered and we write

$$E_1 = \frac{y_{N+1} - y_N}{y_N - y_{N-1}}, \quad G_1 = \left|\frac{f_N - f_p}{f_{N-1} - f_p}\right|^{E_1}, \quad E_2 = \frac{y_N - y_{N-1}}{y_{N-1} - y_{N-2}}, \quad G_2 = \left|\frac{f_{N-1} - f_p}{f_{N-2} - f_p}\right|^{E_2}$$

the two nonlinear boundary conditions consistent with (1) read:

$$f_{N+1} = f_p + (f_N - f_p)G_1, \qquad f_N = f_p + (f_{N-1} - f_p)G_2$$

The Newtonized forms of these read

$$\phi_{N+1} - a\phi_N - b\phi_{N-1} = f_p + (f_N - f_p)G_1 - f_{N-1}$$

$$\phi_N - c\phi_{N-1} - d\phi_{N-2} = f_p + (f_{N-1} - f_p)G_2 - f_N$$

where

$$a = (1 + E_1)G_1, \qquad b = -E_1G_1(f_N - f_p)/(f_{N-1} - f_p)$$

$$c = (1 + E_2)G_2, \qquad d = -E_2G_2(f_{N-1} - f_p)/(f_{N-2} - f_p)$$

These nonlinear boundary conditions have now been implemented in the biharmonic code and compared with simpler downstream boundary conditions for the sudden expansion problem. The results in the table below show the decay of the centerline velocity for a 21 x 31 grid with $y_{max} = 3.0$ and for the same grid with $y_{max} = 2.46, 1.94, 1.48$ and 1.07, i.e. with the asymptotic conditions imposed at $j = 28, 25, 22$ and 19 instead of $j = 31$. For the nonlinear boundary conditions the $f_p(x)$ distribution used was the one appropriate to the grid, i.e. it was obtained by solving $f_{xxxx} = 0$ on the given x grid. This is the approximation to the fully-developed profile that one would expect the computed profiles to tend to as $y \to \infty$. For the simpler downstream conditions we used $f_{yy} = f_{yyy} = 0$.

The comparison shows that for graphical accuracy it is quite adequate to impose the nonlinear conditions at $y = 1.48$ instead of $y = 3.0$; thus comparing values at $x = 1.6$ with those obtained for $x_{max} = 3.0$, we see that the nonlinear conditions give an error of 0.0085 whereas for the simpler condition the error is about -0.046.

One may also observe from these results the rate at which the effect of the choice of boundary condition decays upstream. For example, if either condition is imposed at $x = 1.48$, the difference could hardly be detected to graphical accuracy at $x = 1.07$.

Runs so far made on this problem have been mainly with $R = 41.3$ and have started from an initial guess consisting of a linear interpolation between the inlet conditions and the fully developed downstream profile. The iteration strategy has normally been a maximum of 2 Newtons followed by "Chords"[1] and the monitored iterative corrections printed out were, for the finest grids considered so far (21 x 61), for example,

$$0.19,\ 0.34 \times 10^{-1},\ -0.31 \times 10^{-2},\ 0.64 \times 10^{-3},\ -0.72 \times 10^{-4},\ 0.79 \times 10^{-5}$$

Thus here 2 Newtons + 4 Chords were required to achieve the required accuracy of $1/2 \times 10^{-4}$ in f, and this took about 0.98 minutes CPU time on the IBM 370.

The development of the longitudinal velocity profiles from the inlet conditions to the downstream fully-developed profile is shown in Fig. 6 for the 21 x 31 grid in comparison with Durst et al. experimental data[4]. For the present calculation, the experimental data at the inlet was served as the initial profiles, which were far from the fully developed and of which the corresponding Reynolds number in our notation was 37.3 instead of 41.3 for the corresponding fully-developed inlet conditions. The agreement between the calculated and experimental results is very good; a small discrepancy of the velocity at the centerline in the inlet region could be caused by the inaccuracy in reading the experimental data, particularly at the inlet which was plotted in a small-scale figure.

Centerline Velocities with Various Downstream Conditions (R = 41.3)

x	exp	BL	exp	BL	exp	BL	exp	BL
0.637	3.0856	3.0856	3.0857	3.0857	3.0857	3.0856	3.0857	3.0856
0.733	2.9283	2.9283	2.9284	2.9284	2.9284	2.9284	2.9284	2.9283
0.837	2.7667	2.7667	2.7667	2.7667	2.7668	2.7668	2.7668	2.7665
0.949	2.6046	2.6047	2.6047	2.6047	2.6047	2.6047	2.6048	2.6040
1.070	2.4465	2.4465	2.4464	2.4465	2.4465	2.4465	2.4468	2.4451
1.198	2.2963	2.2964	2.2963	2.2963	2.2964	2.2963	2.2973	2.2913
1.333	2.1576	2.1578	2.1577	2.1577	2.1578	2.1573	2.1600	2.1472
1.476	2.0333	2.0334	2.0334	2.0334	2.0334	2.0327	2.0407	1.9955
1.626	1.9248	1.9249	1.9249	1.2950	1.9252	1.9211		
1.782	1.8327	1.8328	1.8328	1.8326	1.8331	1.8254		
1.944	1.7564	1.7564	1.7564	1.7562	1.7585	1.7262		
2.112	1.6946	1.6947	1.6947	1.6927				
2.284	1.6458	1.6457	1.6457	1.6424				
2.459	1.6079	1.6078	1.6082	1.5910				
2.638	1.5790	1.5781						
2.818	1.5574	1.5561						
3.000	1.5414	1.5340						

●	□	▲	△	■	○	measured
0.0	0.5833	0.9167	1.250	1.750	3.417	x/D
0.0	0.5833	0.9167	1.250	1.750	3.00	calculated

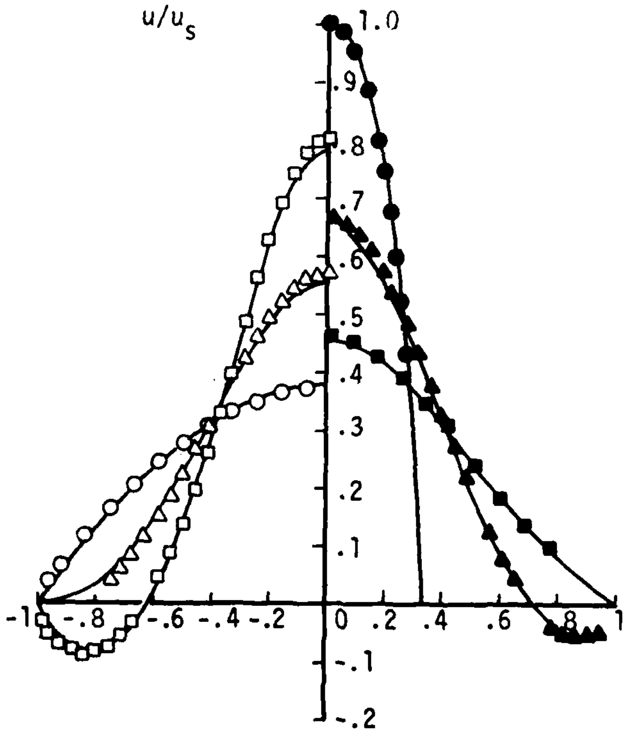

Figure 6. Computed laminar velocity profiles in a sudden expansion: comparison with measurements of Durst et al. (1974).

9.0 TURBULENT FLOW APPLICATIONS

Because of the rapid variation near the wall, we replace the no-slip condition at a suitably small distance from the wall. The law of the wall is the well-known log law

$$\frac{u}{u_\tau} = \frac{1}{\kappa} \log \left(\frac{u_\tau}{\nu} y\right) + c \qquad (9.1)$$

where $u_\tau = \sqrt{\tau_w}/\rho$ and κ and c are appropriate constants. In the law-of-the wall region $50\nu/u_\tau < y < 0.1\delta$ we have

$$\frac{\partial u}{\partial y} = \frac{u_\tau}{\kappa} \cdot \frac{1}{y} \qquad \text{or} \qquad u_\tau = \kappa y \frac{\partial u}{\partial y} \qquad (9.2)$$

So, eliminating u_τ, we have

$$u = y \frac{\partial u}{\partial y} \log \left[\frac{\kappa}{\nu} e^{\kappa c} y^2 \frac{\partial u}{\partial y}\right] \qquad (9.3)$$

If all variables are nondimensionalized this becomes

$$u = y \frac{\partial u}{\partial y} \log \left(\alpha R y^2 \frac{\partial u}{\partial y}\right) \qquad (9.4)$$

where $\alpha = \kappa e^{\kappa c} \simeq 3.3$ and $y > 50/R_w$, where $R_w = u_\tau \ell/\nu$. If we define the function W by

$$W(t) = yt \log (\alpha R y^2 t) \qquad (9.5)$$

the law of the wall in terms of the stream function F reads

$$F_y = W(F_{yy}) \qquad (9.6)$$

In terms of the variable grid difference formulation with $y = Y(1)$ we have

$$b_2 f_0 + b_3 f_1 + b_4 f_2 = W(d_2 f_0 + d_3 f_1 + d_4 f_2) \qquad (9.7)$$

If we wish to express f_0 in terms of f_1 and f_2, as would be required to express the boundary condition in the form (4.1), we use Newton's method again and from an approximation $\bar{f}_0$ determine a better one by

$$f_0 = \bar{f}_0 - \frac{b_2 \bar{f}_0 + b_3 f_1 + b_4 f_2 - W(d_2 \bar{f}_0 + d_3 f_1 + d_4 f_2)}{b_2 - d_2 W'(d_2 \bar{f}_0 + d_3 f_1 + d_4 f_2)} \qquad (9.8)$$

where

$$W'(t) = y[\log(\alpha R y^2 t) + 1]$$

We also need the derivatives $\partial f_o/\partial f_1$ and $\partial f_o/\partial f_2$. These are immediately found by differentiating (9.7) to be

$$\frac{\partial f_o}{\partial f_1} = -\frac{b_3 - d_3 W'}{b_2 - d_2 W'} , \qquad \frac{\partial f_o}{\partial f_2} = -\frac{b_4 - d_4 W'}{b_2 - d_2 W'} \tag{9.9}$$

These provide the values for a and b in (4.2).

A sample of results obtained for developing two-dimensional, plane channel flow is presented on Fig. 7. The Reynolds number corresponds to experimental results of Comte-Bellot[5] and calculated results may be compared with measurements at values of x/D from 20, where the measurements were used as initial conditions for the calculations, to 59. The calculated results deviate from measurements, particularly in the near-wall region, as the flow develops from the initial condition and tends towards the measurements with further downstream distance. The discrepancies are undoubtedly related to the specification of zero cross-stream velocity, in conjunction with the law of the wall, at the initial station. This leads to mass continuity not being initially satisfied and some distance downstream is required before the consequences disappear. This erroneous assumption can be removed with some effort but this was not done here since the main purpose was to develop and evaluate numerical aspects of the code.

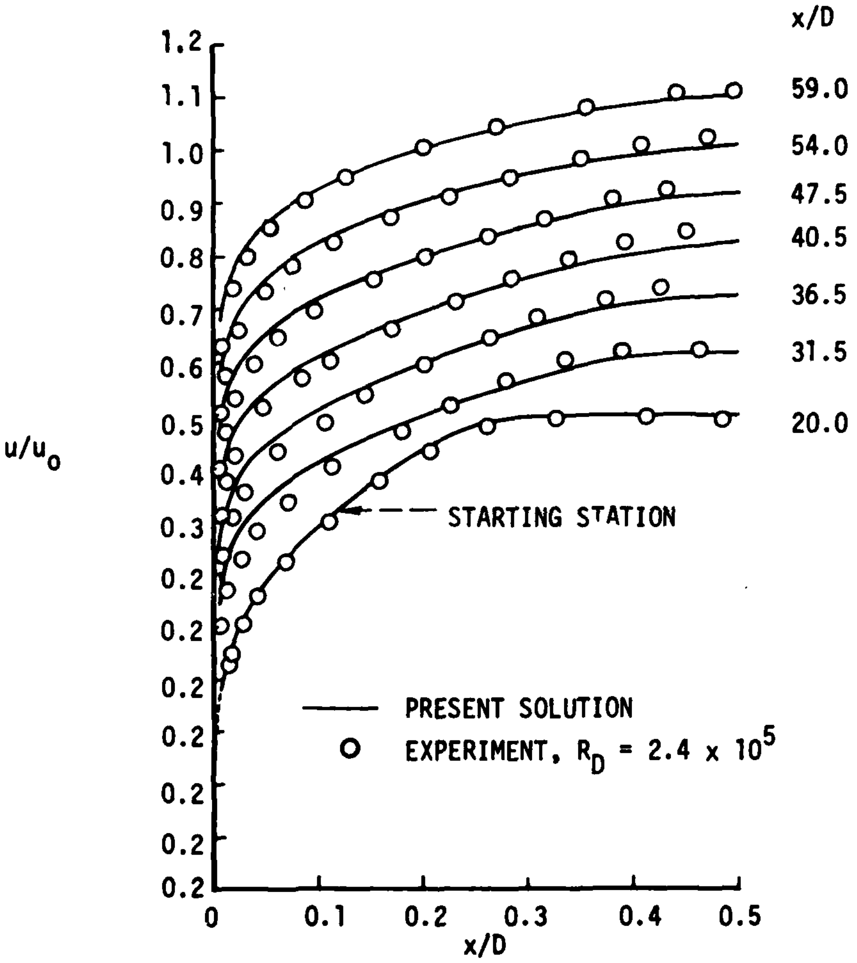

Figure 7. Computed turbulent velocity profiles in a plane channel; comparison with measurements of Comte-Bellot (1968).

10.0 REFERENCES

1. Cebeci, T., Hirsh, R.S., Keller, H.B. and Williams, P.G.: Studies of Numerical Methods for the Plane Navier-Stokes Equation. Report No. MDC J8525, 1979.

2. Isaacson, E. and Keller, H.B.: Analysis of Numerical Methods. Wiley, 1966.

3. Wilson, S.: The Development of Poiseuille Flow. JFM, Vol. 38, 1969, pp. 793-806.

4. Durst, F., Melling, A. and Whitelaw, J.H.: Low Reynolds Number Flow Over a Plane Symmetric Sudden Expansion. JFM, Vol. 64, 1974, pp. 111-128.

5. Comte-Bellot, G.: Ecoulement turbulent entre deux parois paralleles. Publ. Sci. et Tech. du Ministere de l'Air, No. 419, 1965.

Appendix A
BIHARMONIC ADI

Conte (1958) has described an ADI technique for solving the steady biharmonic equation with boundary condition corresponding to the bending of a plate without clamped edges. If $\nabla^4\phi = 0$ is replaced by difference equations in the form

$$(X + Y + Z)\phi = 0$$

where $X\phi$ replaces ϕ_{xxxx}, $Y\phi$ replaces ϕ_{yyyy} and $Z\phi$ replaces $2\phi_{xxyy}$ and the boundary conditions have been used to eliminate exterior ϕ values, then Conte's scheme can be written

$$\hat{\phi} = \phi^{(k)} - \alpha_{k+1}(X\hat{\phi} + Y\phi^{(k)} + Z\phi^{(k)})$$

$$\phi^{(k+1)} = \hat{\phi} - \alpha_{k+1}(Y\phi^{(k+1)} - Y\phi^{(k)})$$

where α_{k+1} is an iteration parameter which can be chosen to accelerate convergence. Conte proves convergence and shows that a good choice of cyclic values for a 20×20 grid is

$$\alpha_k = (0.2)^{1-k}/16 \qquad (k = 1, 2, \ldots, 8)$$

We are interested in first normal derivative boundary conditions (clamped edge) rather than 2nd normal derivative conditions (unclamped edge), but the same scheme can be tried. However, no convergence results appear to be available.

If we drop the k index and denote the new ϕ by ϕ^* and also introduce a nonzero right-hand side, the above scheme can be written in a form suitable for computation as

$$(I + \alpha X)\hat{\phi} = \phi - \alpha[(Y + Z)\phi - r]$$

$$(I + \alpha Y)(\phi^* - \phi) = \hat{\phi} - \phi$$

Each stage here requires sweeps involving the solution of a quin-diagonal system. A subroutine QUISOL was written to solve a general quin-diagonal system of the form

$$a_j\phi_{j-2} + b_j\phi_{j-1} + c_j\phi_j + d_j\phi_{j+1} + e_j\phi_{j+2} = r_j \qquad (j = 1, \ldots, n)$$

with $a_1 = a_2 = b_1 = 0$, $d_n = e_n = e_{n-1} = 0$. This was used to implement the above scheme with a view to using it to solve approximately the Newton equations of section 2 with very little storage requirements. It was tested on a 6 x 6 system whose right-hand sides were generated as the row sums of the coefficients, so that the exact solution was known, namely $\phi_{ij} = 1$. Convergence was excellent and the exact solution obtained, but when the the dimensions were increased to 11 x 11, convergence became very slow and adjustment of the cyclic iteration parameter did not have much effect. The kind of convergence that Conte predicted was not obtained at all, so we conclude that the chanqe of boundary condition makes a vital difference, and that some modification to the scheme would certainly be needed.

APPENDIX B
BOX ADI SCHEME

The Box ADI scheme was originally proposed for dealing with the coupled elliptic equations which occur in corner boundary layers. Since these reduce essentially to the two-dimensional boundary-layer equations at large distances from the corner the view was taken that it would be generally beneficial to construct a scheme that would reduce to the familiar two-dimensional Box scheme far away from the corner. A fully three-dimensional Box scheme has several disadvantages, not least of which is the proliferation of dependent variables, so we proceed as though a standard ADI scheme based on central differences was intended then use the ordinary Box scheme for the boundary value problems that have to be solved on each grid line. This means that slightly different difference approximations are being used in each direction, but they are all second order so this should not matter too much; in fact there is one way in which it may be an advantage, which is best explained after the details of the scheme have been described.

We considered first a simple example to test the scheme, namely

$$u_{xx} + u_{yy} + au_x + bu_y + cu = f(x,y) + \alpha u_t$$

where $f(x,y)$ was chosen so that the equation had a given known solution and appropriate boundary values were imposed as a rectangle. A grid which may be nonuniform in each direction, is set up to cover the rectangle as in Fig. 3.

In the standard ADI scheme we first consider sweeps in the x direction and determine $\hat{u}$ at $t + \frac{1}{2}\Delta t$ to satisfy

$$\hat{u}_{xx} + a\hat{u}_x + \left(\frac{1}{2}c - \frac{2\alpha}{\Delta t}\right)\hat{u} = f(x,y) - u_{yy} - b_{yy} - \left(\frac{1}{2}c + \frac{2\alpha}{\Delta t}\right)u \qquad (B1)$$

on each x grid line $j = 2, \ldots, n-1$, where u is the known field at the previous time step t. We now sweep in the y-direction to solve for $\hat{\hat{u}}$

$$\hat{\hat{u}}_{yy} + b\hat{\hat{u}}_y + \left(\frac{1}{2}c - \frac{2\alpha}{\Delta t}\right)\hat{\hat{u}} = f(x,y) - \hat{u}_{xx} - u\hat{u}_x - \left(\frac{1}{2}c + \frac{2\alpha}{\Delta t}\right)\hat{u} \qquad (B2)$$

on each y-grid line $i = 2, \ldots, n-1$. We then take $\hat{\hat{u}}$ as the updated field for $t + \Delta t$. Since we are interested in the steady state, we repeat these sweeps until convergence.

In the standard ADI scheme we solve the boundary value problems associated with (B1) and (B2) by central differences and a tridiagonal solver, with the right-hand side derivatives evaluated by central differences. In the Box ADI scheme we solve them by the Box method and a SOLV2 subroutine, but still evaluate the right-hand side derivatives by central differences. For (B1), for example, we introduce $v = \hat{u}_x$ and consider the first-order system

$$\begin{aligned} &\hat{u}_x - v = 0 \\ &v_x + av + (\tfrac{c}{2} - \tfrac{2\alpha}{\Delta t})\hat{u} = f(x,y) - u^{**} - bu^{*} - (\tfrac{c}{2} + \tfrac{2\alpha}{\Delta t})u \end{aligned} \tag{B3}$$

where u^{**}, u^{*} are u_{xx}, u_x evaluated by central differences.

If $h_i = x_{i+1} - x_i$, then the Box scheme approximations yield for $i = 2, \ldots, m$

$$\begin{aligned} &\frac{1}{h_{i-1}}(\hat{u}_i - \hat{u}_{i-1}) - \frac{1}{2}(v_i + v_{i-1}) = 0 \\ &\frac{1}{h_{i-1}}(v_i - v_{i-1}) + \frac{a}{2}(v_i + v_{i-1}) + (\frac{c}{4} - \frac{4\alpha}{\Delta t})(\hat{u}_i + \hat{u}_{i-1}) = r_i, \end{aligned} \tag{B4}$$

where

$$r_i = \frac{1}{2}[f(x_{i,y}) + f(x_{i-1,y})] - \frac{1}{2}(u_i^{**} + u_{i-1}^{**}) - \frac{b}{2}(u_i^{*} + u_{i-1}^{*}) - \frac{1}{2}(\frac{c}{2} + \frac{2\alpha}{\Delta t})(u_i + u_{i-1})$$

These equations, together with the boundary conditions, are solved either by a block tri-diagonal solver or a band solver. We can overwrite the u on the $\hat{u}$ provided we have stored temporarily the old u's needed to calculate the u^{**}, u^{*} for the next grid line.

Having performed those calculations for each x grid line, i.e., $j = 2, \ldots, n-1$, we change to sweeping in the *y*-direction. We can use v again, but now for $\hat{\hat{u}}_y$. The system we now solve along each x grid line is the same as (B3) except that the subscript x becomes subscript y, a and b are interchanged and u^{**}, u^{*} now mean derivatives in the x-direction evaluated by central differences. Further we interpret $\hat{u}$ as $\hat{\hat{u}}$ and u and $\hat{u}$, which is natural anyway since u will have been overwritten. The Box equations for the *y-direction follow by similar interpretations* and also changing i to j and interpreting h_j as $y_{j+1} - y_j$.

One might expect that, once the iterations had converged to a level corresponding to the differences between the two approximation schemes, the iterative changes would not decrease further and that this level would therefore give an indication of the truncation error; thus we would have an automatic criterion for terminating the ADI iterations. However, on the simple examples tested the iteration seemed to converge to normal working accuracy, approximately six significant figures.

When the scheme was tried on the vorticity/stream-function formulation of the Navier-Stokes equations, the situation was not so satisfactory. We now have two coupled equations, of course, and moreover we have two boundary conditions on one variable but none on the other, so the situation is somewhat different. However, it was hoped that by solving the two equations simultaneously on each grid line by the Box method, the boundary condition problem would be automatically dealt with.

If f is the stream function and q the vorticity, the equations read

$$\begin{aligned} f_{xx} + f_{yy} &= q + \alpha f_t \\ \frac{1}{R}(q_{xx} + q_{yy}) &= f_y q_x - f_x q_y + q_t \end{aligned} \tag{B5}$$

where αf_t is a fictitious time-dependent term introduced to help convergence, if necessary.

For the x-sweep equations we introduce $g = \hat{f}_x$ and $s = \hat{q}_x$, and the equations corresponding to (B3) then read

$$\begin{aligned} \hat{f}_x - g &= 0 \\ g_x - \frac{2\alpha}{\Delta t}\hat{f} - \frac{1}{2}\hat{q} &= -\frac{2\alpha}{\Delta t} f - f^{**} + \frac{1}{2} q \\ \hat{q}_x - s &= 0 \\ s_x - (Rf_y)s - \frac{2R}{\Delta t} q + (Rq^*)g &= -\frac{2R}{\Delta t} q - q^{**} \end{aligned} \tag{B6}$$

These are discretized by the Box scheme and solved by a block tri-diagonal solver or a band solver in exactly the same way as before, and similarly for the y-sweep equations. Note that we do not require initial guesses for g and s because they do not occur as coefficients or in the right-hand side.

Successive pairs of alternate sweeps using this scheme should gradually update the f and q fields to the solutions of the steady-state equations. In fact, this occurred, when the scheme was tried on the simple Poiseuille flow with the initial guess taken to be the exact solution plus a perturbation which was zero at the corners.

Now there is a slight difficulty at the corners. In the x-sweeps we update the vorticity on the whole of each grid line for $j = 2, \ldots, n-1$. Thus we obtain updated q values along the left- and right-hand boundaries, but not at the corners. When we proceed to the y-sweeps, we update q along the top and bottom boundaries, but not at the corners. Thus some other means must be found for updating the corner vorticity values since they are needed to evaluate, for example, the q^{**} boundary values. We chose to update the value of q at a corner by taking the average of the two values obtained by extrapolating quadratically along each of the adjacent sides. With this procedure the iterations converged for the simple Poiseuille flow, even when the perturbation from the exact solution was not zero at the corners. It was found that the fictitious time derivative term in (B5) was not needed, so α was set to zero.

When the scheme was applied to the channel inlet problem with a uniform parallel entry velocity, no convergence could be obtained at all. Here the initial guess chosen was fully developed flow everywhere except right at the inlet, where the entry conditions were imposed. Because there are infinite vorticity values at the corners in this case, it seems highly likely that it is the corner problem that is affecting the rest of the solution. We hope to try the scheme on a more realistic problem with infinite vorticity values as a part of the future program.